

DECENTRALIZED SUPERVISORY CONTROL OF NETWORKS OF NONLINEAR CONTROL SYSTEMS

GIORDANO POLA¹, PIERDOMENICO PEPE¹ AND MARIA D. DI BENEDETTO¹

ABSTRACT. The paradigm of Cyber–Physical Systems of Systems (CPSoS) is becoming rather popular in the control systems research community because of its expressive power able to properly handle many engineered complex systems of interest. Decentralized control techniques offer a promising approach in taming the inherent complexity of CPSoS, also connected with the design of needed communication infrastructures and computing units. In this paper, we propose decentralized control of networks of discrete–time nonlinear control systems, enforcing complex specifications expressed in terms of regular languages, within any desired accuracy. As discussed in the paper, regular languages, while being traditionally studied in the research community of discrete–event systems, also provide a useful mean to model a rather wide variety of complex specifications for control systems. The design of decentralized controllers is based on formal methods and in particular, on the use of discrete abstractions. Efficient synthesis of such controllers is derived by resorting to on–the–fly algorithmic techniques that also allow the use of parallel computing architectures. Advantages and disadvantages of the decentralized approach over a centralized one, also in terms of computational complexity, are discussed. An illustrative example is presented, which shows the applicability and effectiveness of the results proposed.

1. INTRODUCTION

The novel paradigm of Cyber–Physical Systems of Systems (CPSoS) offers a solid framework where to model, analyze and design complex systems arising in diverse application domains of interest, as for example, smart transportation and logistics, smart power systems and smart buildings, efficient industrial production and gas, water etc. networks. Following [12], CPSoS are characterized by large, often spatially distributed physical systems with complex dynamics, distributed control, supervision and management, partial autonomy of the subsystems, dynamic reconfiguration of the overall system on different time–scales, possibility of emerging behaviors, continuous evolution of the overall system during its operation.

Decentralized control techniques offer a promising approach in taming the inherent complexity of CPSoS. Main advantages of decentralized control architectures over the centralized ones are: (i) they are effective in cases where full state of possibly spatially distributed plants cannot be accessed by a centralized controller, because of potential physical constraints; (ii) they require no communication infrastructures and limited computing units resources with respect to centralized architectures; (iii) scalability, decentralized control architectures are often suited to control large–scale and distributed plants.

Several decentralized control techniques have been proposed in diverse research areas ranging from e.g. decentralized stabilization and regulation [11, 26], robust stabilization, optimization, reliability design [27, 24, 16], decentralized adaptive control [20], consensus and formation control problems in multi–agent systems [17, 19, 18], with e.g. application to mobile robotics [15, 5], to decentralized supervisory control of discrete–event systems (DES), e.g. [25, 6]. In particular, the last research area offers a systematic approach to enforce complex specifications expressed in terms of regular languages on large–scale *qualitative* systems as DES are.

The aim of this paper is to transfer this decentralized control design methodology from *qualitative* systems to *quantitative* systems, described by a network of discrete–time nonlinear (infinite states) control systems Σ_i . Decentralized control architecture consists of a collection of local controllers C_i , each one associated with Σ_i . As also generally assumed in decentralized control of dynamical systems, local controllers C_i are not allowed

The research leading to these results has been partially supported by the Center of Excellence DEWS..

to communicate. We focus on specifications expressed as regular languages, traditionally considered in the control design of DES. This class, when used in the control design of purely continuous (or hybrid) systems, is rather rich and, as also pointed out in [28], comprises reachability and motion planning specifications, periodic orbits, state-based switching specifications, specifications involving sequences of smaller tasks that need to be performed according to a given order. Moreover, operators known for regular languages, and for automata recognizing them, as for example concatenation, union, intersection and complement, see e.g. [6], provide a useful mean assisting the designer in properly modeling desired complex specifications. The approach that we use to solve our decentralized control problem is based on formal methods, see e.g. [8], and in particular, on the use of discrete abstractions, also called symbolic models, see e.g. [1, 29]. Symbolic models are abstract and *finite* descriptions of *infinite states* control systems where each state corresponds to an aggregate of continuous states and each label to an aggregate of control inputs. In this regard, the recent work [23], proposing networks of symbolic models approximating networks of discrete-time nonlinear control systems, provides a useful framework where to solve our decentralized control problem. Indeed, once networks of symbolic models have been constructed, one can design decentralized controllers for purely symbolic/discrete processes, as e.g. DES are. However, current methodologies known for decentralized control of DES cannot be used in our framework because, as also recalled in the paper through an example, local controllers in the DES domain contribute *sequentially* in enforcing the global specification, whereas in our framework local controllers contribute *concurrently* in enforcing the global specification. This key difference asks for classical results available for DES to be revisited. This is the object of investigation of the present paper. We design local controllers enforcing a given regular language specification on the original network of control systems, within any desired accuracy. By following the general ideas of on-the-fly algorithms as in e.g. [30, 9], and in particular, by extending [21] to a decentralized setting, we propose efficient controllers synthesis that can be also implemented via parallel computing architectures. A comparison with a centralized approach is formally discussed, which shows that the parts of the specification that can be enforced through a centralized control architecture and through a decentralized control architecture coincide; the only limitation of decentralized architectures lies in the need for local controllers to agree in advance on which part of the specification to enforce, which however, as discussed in the paper also through an example, is intrinsic of any decentralized control architecture not allowing local controllers to communicate. This is important because, even when physical constraints allow designing communication and computing infrastructures needed in centralized control architectures, one can use decentralized control architectures, thus saving resources for designing and effectively implementing needed infrastructures. Advantages in terms of computational complexity of proposed decentralized controllers over to the centralized ones are discussed. An illustrative example is also included, which shows the applicability and effectiveness of the results proposed.

To the best of our knowledge, formal methods techniques developed in this paper have not yet been explored in the current literature on decentralized control of dynamical systems, with the only exception of [3] which however, is tailored on special classes of regular language specifications.

Preliminary results of this paper are reported in [22]. This paper extends [22] by providing novel results on on-the-fly algorithms, a formal comparison with centralized control architectures and an illustrative example. The paper is organized as follows. Section 2 introduces the decentralized control problem set-up. Section 3 introduces some preliminary results that are used in Section 4 to derive the solution to the control problem. Formal comparison with centralized control architectures is discussed in Section 5. Section 6 presents on-the-fly algorithms and derives computational complexity analysis. Section 7 offers an illustrative example. Concluding remarks are given in Section 8. Notation and basic definitions are reported in the Appendix.

2. NETWORKS OF CONTROL SYSTEMS AND PROBLEM FORMULATION

The network of control systems that we consider in this paper is given as the interconnection of the discrete-time nonlinear control systems $\Sigma_1, \Sigma_2, \dots, \Sigma_N$ described by:

$$(2.1) \quad \Sigma_i : \begin{cases} x_i(t+1) = f_i(x_1(t), \dots, x_N(t), u_i(t)), \\ x_i(t) \in \mathbb{R}^{n_i}, u_i(t) \in \mathcal{U}_i \subset \mathbb{R}^{m_i}, t \in \mathbb{N}_0. \end{cases}$$

Let $n = \sum_{i \in [1;N]} n_i$ and $m = \sum_{i \in [1;N]} m_i$. Functions $f_i : \mathbb{R}^n \times \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_i}$ are assumed to be continuous in their arguments and satisfying $f_i(0_n, 0_{m_i}) = 0_{n_i}$. Sets \mathcal{U}_i are assumed to be finite and containing the origin 0_{m_i} ; this assumption is motivated by concrete applications in e.g. CPSoS where control inputs can only assume a finite number of values. A trajectory of Σ_i is a function

$$x_i : [0; t_{f,i}] \rightarrow \mathbb{R}^{n_i}$$

satisfying (2.1) for all times $t \in [0; t_{f,i}[$, for some time $t_{f,i} \in \mathbb{N}_0$, where we set $[0; 0[= \emptyset$. For coincide notation, we may also refer to the network of control systems in (2.1) by the control system

$$(2.2) \quad \Sigma : \begin{cases} x(t+1) = f(x(t), u(t)), \\ x(t) \in \mathbb{R}^n, u(t) \in \mathcal{U} \subset \mathbb{R}^m, t \in \mathbb{N}_0, \end{cases}$$

where $\mathcal{U} = \times_{i \in [1;N]} \mathcal{U}_i$ and $f(x, (u_1, \dots, u_N)) = (f_1(x, u_1), \dots, f_N(x, u_N))$ for any $x \in \mathbb{R}^n$ and $(u_1, \dots, u_N) \in \mathbb{R}^m$. A trajectory of Σ is a function

$$(2.3) \quad x : [0; t_f] \rightarrow \mathbb{R}^n$$

satisfying (2.2) for all times $t \in [0; t_f[$, for some time $t_f \in \mathbb{N}_0$. Trajectory

$$(2.4) \quad x' : [0; t'_f] \rightarrow \mathbb{R}^n$$

is said to be a continuation of trajectory $x(\cdot)$ as in (2.3), if $t_f < t'_f$ and $x'(t) = x(t)$ for all $t \in [0; t_f]$.

We now formalize the class of specifications we focus on in this paper. Let Y_Q be a finite subset of the state space \mathbb{R}^n of Σ . The specification is expressed as a regular language

$$(2.5) \quad L_Q \subset Y_Q^*,$$

where Y_Q^* is the Kleene closure of Y_Q . This class of specifications is rather rich and comprises, as also pointed out in [28], reachability and motion planning specifications, periodic orbits, state-based switching specifications, specifications involving sequences of smaller tasks that need to be performed according to a given order.

We now define the class of decentralized controllers we consider. To this purpose, consider the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, describing the interaction among subsystems Σ_i in the network, where $\mathcal{V} = [1; N]$ and $(j, i) \in \mathcal{E}$, if function f_i of Σ_i depends explicitly on variable x_j or equivalently, there exist $y_j, z_j \in \mathbb{R}^{n_j}$ such that $f_i(x_1, \dots, x_{j-1}, y_j, x_{j+1}, \dots, x_n, u_i) \neq f_i(x_1, \dots, x_{j-1}, z_j, x_{j+1}, \dots, x_n, u_i)$ for some $x_k \in \mathbb{R}^{n_k}$, $k \in [1; N]$ and $k \neq j$, and $u_i \in \mathcal{U}_i$. Moreover, for any $i \in [1; N]$ define

$$(2.6) \quad \mathcal{N}(i) = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}.$$

For later purposes, for any $i \in [1; N]$ consider also the functions

$$(2.7) \quad \psi_i : \mathbb{R}^{n_i} \times \left(\bigtimes_{j \in \mathcal{N}(i)} \mathbb{R}^{n_j} \right) \times \mathcal{U}_i \rightarrow \mathbb{R}^{n_i}$$

such that $\psi_i(x_i, w_i, u_i) = f_i(x, u_i)$ with $w_i = (x_{j_1}, x_{j_2}, \dots, x_{j_{l_i}}) \in (\times_{j_s \in \mathcal{N}(i)} \mathbb{R}^{n_{j_s}})$, for all $u_i \in \mathcal{U}_i$ and all $x = (x_1, x_2, \dots, x_N) \in \mathbb{R}^n$. We assume a decentralized architecture for the controller which is then specified as a collection of local dynamic controllers C_i , where C_i is associated with Σ_i , in the form of

$$(2.8) \quad C_i : \begin{cases} x_{c,i}(t+1) = f_{c,i}(x_{c,i}(t)), \\ u_i(t) \in h_{c,i}(x_{c,i}(t)) \subseteq \mathcal{U}_i, \\ x_{c,i}(0) \in X_{c,i}^0, \\ x_{c,i}(t) \in X_{c,i}, t \in \mathbb{N}_0, \end{cases}$$

where $x_{c,i}(t)$ is the state of C_i and $u_i(t)$ is the output of C_i at time t . Controllers C_i are open-loop, i.e. they do not depend on the current states $x_i(t)$ and $x_j(t)$ with $j \in \mathcal{N}(i)$, as instead often assumed in decentralized control of dynamical systems. We defer to Remarks 4.9 and 5.6 a discussion in this regard. Let

$$C = (C_1, C_2, \dots, C_N)$$

be the decentralized controller applied to the network of control systems Σ_i . Interaction between control systems Σ_i in the network and local controllers C_i is obtained by coupling Eqns. (2.1) and (2.8), for all

$i \in [1; N]$, and denoted as Σ^C . Control system Σ^C may exhibit blocking behaviors. In fact, existence of a trajectory $x(\cdot)$ of Σ^C in the form of (2.3) implicitly requires that for all times $t \in [0; t_f[$ and $i \in [1; N]$:

$$(2.9) \quad h_{c,i}(x_{c,i}(t)) \neq \emptyset.$$

We can now formalize the control problem we consider:

Problem 2.1. *Given the network of control systems Σ_i in (2.1), the regular language specification L_Q in (2.5) and a desired accuracy $\theta \in \mathbb{R}^+$, find a set of initial states $\mathcal{X}_0 \subseteq \mathbb{R}^n$, a set of final states $\mathcal{X}_f \subseteq \mathbb{R}^n$ and a collection of local controllers C_i in (2.8) such that:*

- (i) *for any trajectory $x(\cdot)$ of Σ^C as in (2.3) with $x(0) \in \mathcal{X}_0$, either $x(t_f) \in \mathcal{X}_f$ or there exists a continuation $x'(\cdot)$ of $x(\cdot)$, as in (2.4), such that $x'(t_f) \in \mathcal{X}_f$;*
- (ii) *for any trajectory $x(\cdot)$ of Σ^C as in (2.3) with $x(0) \in \mathcal{X}_0$ and $x(t_f) \in \mathcal{X}_f$, there exists a word $q_0 q_1 \dots q_{t_f} \in L_Q$ such that for all times $t \in [0; t_f]$:*

$$(2.10) \quad \|x(t) - q_t\| \leq \theta.$$

Since condition (2.10) relaxes condition $x(t) = q_t$, Problem 2.1 can be thought of as an approximate version of classical decentralized supervisory control problems traditionally given for DES (see e.g. [6]) and here extended to networks of (infinite states) nonlinear control systems. Further discussion in this regard is reported in Remark 5.6 in Section 5.

For later purposes, we give the following

Definition 2.2. Triplet $(C, \mathcal{X}_0, \mathcal{X}_f)$ is said to enforce a word $q_0 q_1 \dots q_{t_f} \in L_Q$ within accuracy θ on Σ , if there exists a trajectory $x(\cdot)$ of Σ^C as in (2.3) with $x(0) \in \mathcal{X}_0$ and $x(t_f) \in \mathcal{X}_f$ satisfying (2.10) for all times $t \in [0; t_f]$.

3. APPROXIMATING NETWORKS OF CONTROL SYSTEMS

In this section we propose some results based on [23] and concerning the construction of networks of symbolic models approximating networks of control systems. A symbolic model is an abstract description of a control system where each state corresponds to an aggregate of continuous states and each label to an aggregate of control inputs. We start by giving a representation of Σ in terms of systems:

Definition 3.1. Given Σ , define the system

$$S(\Sigma) = (X, X_0, U, \longrightarrow, X_m, Y, H)$$

where $X = X_0 = X_m = \mathbb{R}^n$, $U = \mathcal{U}$, $x \xrightarrow{u} x^+$ if $x^+ = f(x, u)$, $Y = \mathbb{R}^n$, and $H(x) = x$ for any $x \in \mathbb{R}^n$.

System $S(\Sigma)$ is metric when we regard $Y = \mathbb{R}^n$ as being equipped with the metric $\mathbf{d}(x, x') = \max_{i \in [1; N]} \mathbf{d}_i(x_i, x'_i)$, for all $x = (x_1, x_2, \dots, x_N)$, $x' = (x'_1, x'_2, \dots, x'_N) \in \mathbb{R}^n$, where \mathbf{d}_i , defined by $\mathbf{d}_i(x_i, x'_i) = \|x_i - x'_i\|$ for all $x_i, x'_i \in \mathbb{R}^{n_i}$, is the metric used for \mathbb{R}^{n_i} . System $S(\Sigma)$ will be approximated by means of networks of systems that are introduced in the following

Definition 3.2. Given Σ_i , $i \in [1; N]$ and a quantization vector $\eta \in \mathbb{R}_N^+$, define the system

$$S^\eta(\Sigma_i) = (X_i^\eta, X_{i,0}^\eta, W_i^\eta \times U_i^\eta, \xrightarrow[\eta, i]{}, X_{i,m}^\eta, Y_i^\eta, H_i^\eta)$$

where:

- $X_i^\eta = X_{i,0}^\eta = X_{i,m}^\eta = \eta(i)\mathbb{Z}^{n_i}$;
- $W_i^\eta = \times_{j \in \mathcal{N}(i)} \eta(j)\mathbb{Z}^{n_j}$;
- $U_i^\eta = \mathcal{U}_i$;
- $\xi_i \xrightarrow[\eta, i]{(w_i, u_i)} \xi_i^+$, if $\xi_i^+ = [\psi_i(\xi_i, w_i, u_i)]_{\eta(i)}$ with ψ_i in (2.7);
- $Y_i^\eta = \mathbb{R}^{n_i}$;
- $H_i^\eta(\xi_i) = \xi_i$ for any $\xi_i \in X_i^\eta$.

Each system $S^\eta(\Sigma_i)$ approximates each control system Σ_i in the network for any desired accuracy. It is countable and becomes symbolic when one is interested in the dynamics of Σ_i on a bounded subset of \mathbb{R}^{n_i} , as in most applications of interest and also in this paper, see Section 4. System $S^\eta(\Sigma_i)$ is metric with metric \mathbf{d}_i . By definition of the transition relation $\xrightarrow{\eta, i}$ and since operator $[\cdot]_{\eta(i)}$ is a function, system $S^\eta(\Sigma_i)$ is deterministic. Interaction among systems $S^\eta(\Sigma_i)$ is formalized by the following

Definition 3.3. [23] Given $S^\eta(\Sigma_i)$, $i \in [1; N]$, define the network of systems

$$\mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1; N]}) = (X^\eta, X_0^\eta, U^\eta, \xrightarrow{\eta}, X_m^\eta, Y^\eta, H^\eta)$$

where:

- $X^\eta = X_0^\eta = X_m^\eta = \times_{i \in [1; N]} X_i^\eta$;
- $U^\eta = \times_{i \in [1; N]} U_i^\eta$;
- $(\xi_1, \dots, \xi_N) \xrightarrow[\eta]{(u_1, \dots, u_N)} (\xi_1^+, \dots, \xi_N^+)$, if $\xi_i \xrightarrow[\eta, i]{(w_i, u_i)} \xi_i^+$ with $w_i = (\xi_{j_1}, \xi_{j_2}, \dots, \xi_{j_{i_i}})$, $j_s \in \mathcal{N}(i)$ for any $i \in [1; N]$;
- $Y^\eta = \times_{i \in [1; N]} Y_i^\eta$;
- $H^\eta(\xi_1, \dots, \xi_N) = (H_1^\eta(\xi_1), \dots, H_N^\eta(\xi_N))$.

System $\mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1; N]})$ is metric with metric \mathbf{d} and inherits from systems $S^\eta(\Sigma_i)$ the properties of being deterministic and countable/symbolic. In the sequel we consider the following

Assumption 3.4. *There exists a locally Lipschitz, incrementally globally asymptotically stable (δ -GAS) Lyapunov function (see [2])*

$$(3.1) \quad V : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}_0^+$$

for Σ , i.e. function V satisfies the following inequalities for all $x, x' \in \mathbb{R}^n$ and $u \in \mathcal{U}$:

- (i) $\underline{\alpha}(\|x - x'\|) \leq V(x, x') \leq \overline{\alpha}(\|x - x'\|)$,
- (ii) $V(f(x, u), f(x', u)) - V(x, x') \leq -\rho(V(x, x'))$,

for some \mathcal{K}_∞ functions $\underline{\alpha}$, $\overline{\alpha}$, ρ .

Throughout the paper we assume the existence of a \mathcal{K}_∞ function σ such that the δ -GAS Lyapunov function V satisfies for all $x, y, z \in \mathbb{R}^n$

$$(3.2) \quad |V(x, y) - V(x, z)| \leq \sigma(\|y - z\|).$$

The above assumption is not restrictive since in order to solve Problem 2.1 we are interested in the dynamics of Σ on a bounded subset of \mathbb{R}^n (see Section 4). We now have all the ingredients to present the following

Proposition 3.5. *Suppose that Assumption 3.4 holds. Then, for any desired accuracy $\mu \in \mathbb{R}^+$ and for any quantization vector $\eta \in \mathbb{R}_N^+$ satisfying the following inequality*

$$(3.3) \quad \|\eta\| \leq \min \{(\sigma^{-1} \circ \rho \circ \underline{\alpha})(\mu), (\overline{\alpha}^{-1} \circ \underline{\alpha})(\mu)\},$$

relation $\mathcal{R}_\mu \subseteq \mathbb{R}^n \times X^\eta$ specified by

$$(3.4) \quad (x, \xi) \in \mathcal{R}_\mu \Leftrightarrow V(x, \xi) \leq \underline{\alpha}(\mu)$$

is a strong μ -approximate bisimulation between $S(\Sigma)$ and $\mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1; N]})$. Consequently, systems $S(\Sigma)$ and $\mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1; N]})$ are strongly μ -bisimilar.

Proof. Direct consequence of Proposition 1 in [23]. □

The above result requires the existence of a δ -GAS Lyapunov function for Σ . Compositional design of such Lyapunov function can be done by resorting e.g. to the small-gain theorem, see for instance [10]. These arguments have been used in [23] to derive networks of symbolic models approximating networks of discrete-time nonlinear control systems. The main difference between the results reported in this section and in [23] are: (i) while systems in Definition 3.2 are deterministic, those in [23] are nondeterministic; (ii) quantization parameters $\eta(i)$ can be selected here independently from $\eta(j)$ but in the respect of (3.3), while selection of $\eta(i)$ depends on the selection of some other $\eta(j)$ in [23]; (iii) sets \mathcal{U}_i are finite here while they are convex, bounded and with interior in [23].

4. DECENTRALIZED SUPERVISORY CONTROL DESIGN

In this section we provide the solution to Problem 2.1. By using the results in Section 3, the design of decentralized controllers can be translated from a continuous (infinite states) domain to a symbolic (finite) domain. Hence, one could in principle use techniques available for DES to design decentralized controllers, see e.g. [25, 6]. However, these techniques cannot be used in our framework because while local controllers in Problem 2.1 contribute *concurrently* in enforcing the global specification, in decentralized supervisory control of DES, local controllers contribute *sequentially* in enforcing the global specification, as briefly recalled in the following example.

Example 4.1. Consider a finite system S with $\mathcal{L}_m^u(S) = \mathcal{U}^*$ where \mathcal{U}^* denotes the Kleene closure of $\mathcal{U} = \{a, b, c\}$. Consider the regular language specification $L_Q = \{\varepsilon, a, ab, aba, abab, ababa, \dots\}$ where ε is the empty word. Suppose that supervisors (controllers) C_1 and C_2 are characterized by sets of controllable¹ events $\mathcal{U}_1 = \{a, c\}$ and $\mathcal{U}_2 = \{b, c\}$, respectively. Assume further that the set of observable¹ events of C_1 and C_2 coincide in \mathcal{U} . A decentralized control policy enforcing L_Q on S is as follows: supervisor C_1 enforces event a after having measured events b and ε ; supervisor C_2 enforces event b after having measured event a .

Motivated by inherent differences between decentralized control schemes used for DES and in Problem 2.1, we now extend techniques of decentralized supervisory control from DES to our problem set-up. We start with the following

Example 4.2. Consider a network of two control systems Σ_i described by $x_i(t+1) = -2x_i(t) + u_i(t)$, $t \in \mathbb{N}_0$ with $x_i(t) \in \mathbb{R}$, $\mathcal{U}_i = [-1; 1]$ and a specification L_Q described by the collection of words $(0,0)(1,1)$ and $(0,0)(-1,-1)$; set for simplicity the desired accuracy to $\theta = 0$. First of all, corresponding control system Σ satisfies Assumption 3.4. In order for the specification to be enforced by a decentralized controller $C = (C_1, C_2)$, the controllers C_i need to agree on which part of the specification they want to enforce. Indeed, if they want to enforce word $(0,0)(1,1)$, they both select at time $t = 0$ control input $u_i(0) = 1$; instead, if they want to enforce word $(0,0)(-1,-1)$, they both select at time $t = 0$ control input $u_i(0) = -1$. If the controllers C_i do not agree on which word of the specification L_Q to enforce, L_Q cannot be met by using *any* decentralized control architecture. As a matter of fact, if $u_1(0) = 1$ with the purpose of enforcing word $(0,0)(1,1)$, and if $u_2(0) = -1$ with the purpose of enforcing word $(0,0)(-1,-1)$, the state reached at time $t = 1$ is $(1, -1)$ from which, L_Q is not fulfilled.

Remark 4.3. At a general level, the problem raised in the above example can be solved as follows:

(i) *Restriction of the class of specifications.* It is easy to see that the above problem is solved when L_Q is "decoupled", i.e. it can be expressed as $L_Q = L_{Q,1} \times L_{Q,2} \times \dots \times L_{Q,N}$ where each $L_{Q,i}$ is a regular language taking values in the projection of Y_Q^* onto \mathbb{R}^{n_i} and each $L_{Q,i}$ is enforced by a local controller C_i which can be designed independently from any other C_j . This is for instance, the approach taken in [3].

(ii) *Online agreement on the specification word to enforce.* When local controllers are allowed to share information through a to-be designed and implemented communication infrastructure, thus leading to a distributed control architecture, above problem can be solved because controllers can agree online on which word of the specification to enforce.

¹We refer to e.g. [6] for the notions of controllable or observable events.

(iii) *Offline agreement on the specification word to enforce.* When local controllers are not allowed to communicate, as in the decentralized control architecture we consider, controllers can only agree offline and hence in advance on which word of the specification to enforce. The advantage of this approach over the first one is that no restriction on the class of specifications is needed and, over the second approach, is that no communication infrastructure is required. In this paper we follow the third approach.

We suppose that assumption of Proposition 3.5 hold and use strong approximate bisimulation relation \mathcal{R}_μ defined in (3.4). Let the system

$$S'_Q = (X'_Q, X'_{0,Q}, Y_Q, \xrightarrow{\iota, Q}, X'_{Q,m}, Y'_Q, H'_Q)$$

be symbolic, deterministic, accessible and nonblocking and such that its input marked language coincides with the language specification, i.e. $\mathcal{L}_m^u(S'_Q) = L_Q$. Automatic tools for constructing S'_Q are well known in the literature, see e.g. [7]. Given S'_Q , it is useful to define symbolic system S_Q whose states are transitions of S'_Q and vice versa. More formally:

Definition 4.4. Given system S'_Q , define system

$$(4.1) \quad S_Q = (X_Q, X_{Q,0}, U_Q, \xrightarrow{Q}, X_{Q,m}, \mathbb{R}^n, H_Q)$$

where:

- $X_Q = \xrightarrow{\iota, Q}$;
- $X_{Q,0}$ is the collection of states $x'_Q \xrightarrow{\iota, Q} x'^+_Q$ in X_Q with $x'_Q \in X'_{Q,0}$;
- $U_Q = \{u_Q\}$, where u_Q is a dummy input;
- \xrightarrow{Q} is the collection of transitions

$$\left(x^1_Q \xrightarrow{\iota, Q} x^2_Q \right) \xrightarrow{u_Q, Q} \left(x^3_Q \xrightarrow{\iota, Q} x^4_Q \right)$$

with $x^2_Q = x^3_Q$;

- $X_{Q,m}$ is the collection of states $x'_Q \xrightarrow{\iota, Q} x'^+_Q$ in X_Q with $x'^+_Q \in X'_{Q,m}$;
- $H_Q(x'_Q \xrightarrow{\iota, Q} x'^+_Q) = u'_Q$ for any state $x'_Q \xrightarrow{\iota, Q} x'^+_Q$ in X_Q .

From the above definitions it is readily seen that

$$\mathcal{L}^y(S_Q) = \mathcal{L}^u(S'_Q), \quad \mathcal{L}_m^y(S_Q) = \mathcal{L}_m^u(S'_Q) = L_Q.$$

Moreover, S_Q is symbolic, accessible and nonblocking. In the sequel for ease of notation we denote a state $x'_Q \xrightarrow{\iota, Q} x'^+_Q$ of X_Q by x_Q and a transition $x_Q \xrightarrow{u_Q, Q} x^+_Q$ of S_Q by $x_Q \xrightarrow{Q} x^+_Q$. For any $i \in [1; N]$, function

$$H_{Q,i} : X_Q \rightarrow \mathbb{R}^{n_i}$$

denotes the "projection" of function H_Q onto \mathbb{R}^{n_i} , i.e. for all $x_Q \in X_Q$, $H_{Q,i}(x_Q) = q^i$ if $H_Q(x_Q) = (q^1, q^2, \dots, q^N)$. Consider the operators:

$$\begin{aligned} \mathcal{I}_i : \left(\xrightarrow{Q} \right) \times \mathbb{R}_N^+ &\rightarrow \{\text{True}, \text{False}\}, i \in [1; N], \\ \mathcal{I} : \left(\xrightarrow{Q} \right) \times \mathbb{R}_N^+ &\rightarrow \{\text{True}, \text{False}\}. \end{aligned}$$

Consider any $\eta \in \mathbb{R}_N^+$ and any transition $x_Q \xrightarrow{Q} x^+_Q$ in S_Q . Then, for all $i \in [1; N]$ set

$$(4.2) \quad \mathcal{I}_i(x_Q \xrightarrow{Q} x^+_Q, \eta) = \text{True},$$

if there exists a control input $u_i \in U_i^\eta$ of $S^\eta(\Sigma_i)$ such that

$$(4.3) \quad [H_{Q,i}(x_Q)]_{\eta(i)} \xrightarrow[\eta, i]{(v_i, u_i)} [H_{Q,i}(x^+_Q)]_{\eta(i)},$$

where $v_i = ([H_{Q,j_1}(x_Q)]_{\eta(j_1)}, \dots, [H_{Q,j_{l_i}}(x_Q)]_{\eta(j_{l_i})})$, with $j_s \in \mathcal{N}(i)$. If no $u_i \in U_i^\eta$ exists satisfying (4.3), set

$$(4.4) \quad \mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta) = \text{False}.$$

Operator \mathcal{I}_i , when evaluated in $x_Q \xrightarrow{Q} x_Q^+$ and η , is then set to **True** if transition $x_Q \xrightarrow{Q} x_Q^+$ can be matched by the system $S^\eta(\Sigma_i)$ and **False**, otherwise. Since conditions (4.3) involve set of transitions of S_Q and set U_i^η that are finite, operator \mathcal{I}_i can be effectively computed in a finite number of steps. Define:

$$(4.5) \quad \mathcal{I}(x_Q \xrightarrow{Q} x_Q^+, \eta) = \bigwedge_{i \in [1;N]} \mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta).$$

Define the subsystem

$$(4.6) \quad S_{Q,\eta} = (X_Q, X_Q^0, U_Q, \xrightarrow{Q,\eta}, X_{Q,m}, Y_Q, H_Q)$$

of S_Q as in (4.1), where the transition relation $\xrightarrow{Q,\eta} \subseteq \xrightarrow{Q}$ contains all and only transitions $x_Q \xrightarrow{Q} x_Q^+$ of S_Q satisfying the following condition:

$$(4.7) \quad \mathcal{I}(x_Q \xrightarrow{Q} x_Q^+, \eta) = \text{True}.$$

System $S_{Q,\eta}$ captures all transitions of the specification system S_Q that can be matched by the control system Σ^C . However, system $S_{Q,\eta}$ is blocking in general. Since the controllers in Σ^C are required to fulfill condition (2.9), we need to extract from $S_{Q,\eta}$ a subsystem exhibiting nonblocking behavior. This is accomplished by computing the subsystem $\text{Trim}(S_{Q,\eta})$ of $S_{Q,\eta}$ (see Appendix for the definition of Trim) which is indeed nonblocking.

We can now provide the solution to Problem 2.1. We will follow the third approach discussed in Remark 4.3. Consider any word \mathbf{q} marked by $\text{Trim}(S_{Q,\eta})$, i.e. such that $\mathbf{q} \in \mathcal{L}_m^y(\text{Trim}(S_{Q,\eta}))$, and let

$$(4.8) \quad S_{\mathbf{q}} = (X_{\mathbf{q}}, \{x_{\mathbf{q}}^0\}, U_{\mathbf{q}}, \xrightarrow{\mathbf{q}}, \{x_{\mathbf{q},m}\}, Y_{\mathbf{q}}, H_{\mathbf{q}})$$

be a symbolic, accessible and nonblocking system, marking \mathbf{q} , i.e. such that $\mathcal{L}_m^y(S_{\mathbf{q}}) = \{\mathbf{q}\}$. System $S_{\mathbf{q}}$ is characterized by a unique successor of each state. For this reason, in the sequel we write any transition of $S_{\mathbf{q}}$ in the form of $x_{\mathbf{q}} \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^+$ by omitting the corresponding label. For any $i \in [1;N]$, function

$$H_{\mathbf{q},i} : X_{\mathbf{q}} \rightarrow \mathbb{R}^{n_i}$$

denotes the "projection" of function $H_{\mathbf{q}}$ onto \mathbb{R}^{n_i} , i.e. for all $x_{\mathbf{q}} \in X_{\mathbf{q}}$, $H_{\mathbf{q},i}(x_{\mathbf{q}}) = q^i$ if $H_{\mathbf{q}}(x_{\mathbf{q}}) = (q^1, q^2, \dots, q^N)$. Define the following sets:

$$(4.9) \quad \begin{aligned} \mathcal{X}_0 &= \mathcal{R}_\mu^{-1}(\times_{i \in [1;N]} \{[H_{\mathbf{q},i}(x_{\mathbf{q}}^0)]_{\eta(i)}\}), \\ \mathcal{X}_f &= \mathcal{R}_\mu^{-1}(\times_{i \in [1;N]} \{[H_{\mathbf{q},i}(x_{\mathbf{q},m})]_{\eta(i)}\}). \end{aligned}$$

Entities defining C_i in (2.8) are then specified by:

$$(4.10) \quad \begin{aligned} X_{c,i}^0 &= \{x_{\mathbf{q}}^0\}, \\ X_{c,i} &= X_{\mathbf{q}}, \\ f_{c,i}(x_{\mathbf{q}}) &= x_{\mathbf{q}}^+, \text{ if } x_{\mathbf{q}} \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^+, \\ h_{c,i}(x_{\mathbf{q}}) &= \left\{ \begin{array}{l} u_i \in U_i^\eta | x_{\mathbf{q}}^+ = f_{c,i}(x_{\mathbf{q}}) \text{ and} \\ [H_{\mathbf{q},i}(x_{\mathbf{q}})]_{\eta(i)} \xrightarrow{(v_i, u_i)}_{\eta,i} [H_{\mathbf{q},i}(x_{\mathbf{q}}^+)]_{\eta(i)} \end{array} \right\}, \end{aligned}$$

where $v_i = ([H_{\mathbf{q},j_1}(x_{\mathbf{q}})]_{\eta(j_1)}, \dots, [H_{\mathbf{q},j_{l_i}}(x_{\mathbf{q}})]_{\eta(j_{l_i})})$, with $j_s \in \mathcal{N}(i)$.

Remark 4.5. Sets $X_{c,i}^0$, $X_{c,i}$ and function $f_{c,i}$ in (4.10) are the same for all C_i . This feature is essential to solve problems raised in Example 4.2 and discussed in Remark 4.3. As a by-product, this choice has the advantage of requiring limited computational effort that is significant when the number N of subsystems in the network becomes large. We also stress that computation of functions $f_{c,i}$ and $h_{c,i}$ can be done offline, see Section 6, which is important because it reduces online computational time needed by the controllers to ensure timely control action.

We now have all the ingredients to present the main result of this paper.

Theorem 4.6. *Suppose that Assumption 3.4 holds. For any desired accuracy $\theta \in \mathbb{R}^+$ select $\mu \in \mathbb{R}^+$ and $\eta \in \mathbb{R}_N^+$ satisfying (3.3) and*

$$(4.11) \quad \mu + \|\eta\|/2 \leq \theta,$$

Then, sets \mathcal{X}_0 and \mathcal{X}_f in (4.9) and controllers C_i in (2.8) specified by (4.10) solve Problem 2.1.

Proof. Since assumption of Proposition 3.5 holds, by (3.3) we get $S(\Sigma) \cong_\mu \mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1;N]})$; we consider \mathcal{R}_μ in (3.4) as strong μ -approximate bisimulation relation between $S(\Sigma)$ and $\mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1;N]})$. Consider any trajectory $x(\cdot)$ of Σ^C as in (2.3) with initial condition $x(0) = (x_1(0), x_2(0), \dots, x_N(0)) \in \mathcal{X}_0$. Pick $\xi(0) = (\xi_1(0), \xi_2(0), \dots, \xi_N(0))$ such that $\xi_i(0) = [H_{\mathbf{q},i}(x_{\mathbf{q}}^0)]_{\eta(i)}$, $i \in [1;N]$. By definition of \mathcal{X}_0 in (4.9) we get:

$$(4.12) \quad (x(0), \xi(0)) \in \mathcal{R}_\mu.$$

Define $q_0 = H_{\mathbf{q}}(x_{\mathbf{q}}^0)$. By (4.12), definition of \mathcal{R}_μ and of $\xi(0)$ we get

$$\begin{aligned} \|x(0) - \xi(0)\| &\leq \mu, \\ \|\xi(0) - q_0\| &= \|\xi(0) - H_{\mathbf{q}}(x_{\mathbf{q}}^0)\| \\ &= \max_{i \in [1;N]} \|\xi_i(0) - H_{\mathbf{q},i}(x_{\mathbf{q}}^0)\| \\ &\leq \max_{i \in [1;N]} \eta(i)/2 = \|\eta\|/2 \end{aligned}$$

which, combined with (4.11), yields

$$(4.13) \quad \begin{aligned} \|x(0) - q_0\| &\leq \|x(0) - \xi(0)\| + \|\xi(0) - q_0\| \\ &\leq \mu + \|\eta\|/2 \leq \theta. \end{aligned}$$

By the nonblocking property of $S_{\mathbf{q}}$, either (case 1) $x_{\mathbf{q}}^0 \in X_{\mathbf{q},m}$ or (case 2) there exists a transition $x_{\mathbf{q}}^0 \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^1$. In case 1, by (4.12), definition of $\xi(0)$ and (4.9), $x(0) \in \mathcal{X}_f$ and condition (i) of Problem 2.1 holds for $t = t_f = 0$. Moreover, since $q_0 \in L_Q$ and by (4.13), condition (ii) of Problem 2.1 holds as well for $t = t_f = 0$. We now address case 2. By definition of $S_{\mathbf{q}}$, transition $x_{\mathbf{q}}^0 \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^1$ satisfies condition (4.7). Hence, by (4.2) for all $i \in [1;N]$ there exists $u_i(0) \in U_i^\eta$ satisfying

$$(4.14) \quad [H_{\mathbf{q},i}(x_{\mathbf{q}}^0)]_{\eta(i)} \xrightarrow[\eta,i]{(v_i(0), u_i(0))} [H_{\mathbf{q},i}(x_{\mathbf{q}}^1)]_{\eta(i)}$$

where $v_i(0) = ([H_{\mathbf{q},j_1}(x_{\mathbf{q}}^0)]_{\eta(j_1)}, \dots, [H_{\mathbf{q},j_{i-1}}(x_{\mathbf{q}}^0)]_{\eta(j_{i-1})})$, for all $j_s \in \mathcal{N}(i)$. Hence, by definition of $h_{c,i}$ in (4.10), we get $u_i(0) \in h_{c,i}(x_{\mathbf{q}}^0) \neq \emptyset$, for all $i \in [1;N]$, from which, condition (2.9) holds for $t = 0$. Let $u(0) = (u_1(0), u_2(0), \dots, u_N(0))$ and $x(1) = f(x(0), u(0))$. By Definition 3.1 we get:

$$(4.15) \quad x(0) \xrightarrow{u(0)} x(1).$$

Pick $\xi(1) = (\xi_1(1), \xi_2(1), \dots, \xi_N(1))$ such that $\xi_i(1) = [H_{\mathbf{q},i}(x_{\mathbf{q}}^1)]_{\eta(i)}$, $i \in [1;N]$. By (4.14), definitions of $\xi_i(0)$ and $\xi_i(1)$ we get

$$\xi_i(0) \xrightarrow[\eta,i]{(v_i(0), u_i(0))} \xi_i(1),$$

which implies, by definition of $u(0)$ and Definition 3.3

$$(4.16) \quad \xi(0) \xrightarrow[\eta]{u(0)} \xi(1),$$

i.e. the above transition is in $\mathcal{S}(\{S^\eta(\Sigma_i)\}_{i \in [1;N]})$. By (4.15), (4.16), determinism of $S^\eta(\Sigma)$, and definition of \mathcal{R}_μ we get:

$$(x(1), \xi(1)) \in \mathcal{R}_\mu.$$

We now use induction and show that if the following conditions (H1), (H2) and (H3) hold for some $\tau \in \mathbb{N}_0$

(H1) $(x(\tau), \xi(\tau)) \in \mathcal{R}_\mu$, where

$$\begin{aligned} \xi_i(\tau) &= [H_{\mathbf{q},i}(x_{\mathbf{q}}^\tau)]_{\eta(i)}, i \in [1;N], \\ \xi(\tau) &= (\xi_1(\tau), \xi_2(\tau), \dots, \xi_N(\tau)), \end{aligned}$$

$$(H2) \ x_{\mathbf{q}}^0 \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^1 \xrightarrow{\mathbf{q}} \dots \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^{\tau},$$

(H3) Condition (2.10) is satisfied for all $t \in [0; \tau]$ where q_t is defined by $q_t = H_{\mathbf{q}}(x_{\mathbf{q}}^t)$, $t \in [0; \tau]$, then one of the following conditions (T1) or (T2) hold:

(T1) $x(\tau) \in \mathcal{X}_f$ and the word $q_0 q_1 \dots q_{\tau} \in L_Q$;

(T2) condition (2.9) holds for $t = \tau$ and for any $u_i(\tau) \in h_{c,i}(x_{\mathbf{q}}^{\tau})$, $i \in [1; N]$, by setting

$$(4.17) \quad u(\tau) = (u_1(\tau), u_2(\tau), \dots, u_N(\tau)),$$

$$(4.18) \quad x(\tau + 1) = f(x(\tau), u(\tau)),$$

$$(4.19) \quad \xi_i(\tau + 1) = [H_{\mathbf{q},i}(x_{\mathbf{q}}^{\tau+1})]_{\eta(i)}, i \in [1; N],$$

$$(4.20) \quad \xi(\tau + 1) = (\xi_1(\tau + 1), \xi_2(\tau + 1), \dots, \xi_N(\tau + 1)),$$

the following conditions hold:

$$(T2.1) \ (x(\tau + 1), \xi(\tau + 1)) \in \mathcal{R}_{\mu},$$

$$(T2.2) \ x_{\mathbf{q}}^0 \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^1 \xrightarrow{\mathbf{q}} \dots \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^{\tau} \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^{\tau+1},$$

(T2.3) Condition (2.10) is satisfied for all $t \in [0; \tau + 1]$ where q_t is defined by $q_t = H_{\mathbf{q}}(x_{\mathbf{q}}^t)$, $t \in [0; \tau + 1]$.

Let us assume then that (H1)–(H3) hold. By the nonblocking property of $S_{\mathbf{q}}$, either (case 1) $x_{\mathbf{q}}^{\tau} \in X_{\mathbf{q},m}$ or (case 2) there exists a transition $x_{\mathbf{q}}^{\tau} \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^{\tau+1}$. We start by addressing case 1. By (H1) and the definition of \mathcal{X}_f in (4.9) we get $x(\tau) \in \mathcal{X}_f$. Hence, condition (i) of Problem 2.1 holds for $t_f = \tau$. Since $x_{\mathbf{q}}^{\tau} \in X_{\mathbf{q},m}$ then $q_0 q_1 \dots q_{\tau} \in L_Q$ and (T1) is proven. Moreover by (H3), condition (ii) of Problem 2.1 holds for $t_f = \tau$. We now address case 2. First of all (T2.2) holds. By definition of $S_{\mathbf{q}}$, transition $x_{\mathbf{q}}^{\tau} \xrightarrow{\mathbf{q}} x_{\mathbf{q}}^{\tau+1}$ satisfies condition (4.7). Hence, by (4.2), for all $i \in [1; N]$ there exists $u_i(\tau) \in U_i^{\eta}$ satisfying

$$(4.21) \quad [H_{\mathbf{q},i}(x_{\mathbf{q}}^{\tau})]_{\eta(i)} \xrightarrow[\eta,i]{(v_i(\tau), u_i(\tau))} [H_{\mathbf{q},i}(x_{\mathbf{q}}^{\tau+1})]_{\eta(i)}$$

where $v_i(\tau) = ([H_{\mathbf{q},j_1}(x_{\mathbf{q}}^{\tau})]_{\eta(j_1)}, \dots, [H_{\mathbf{q},j_{l_i}}(x_{\mathbf{q}}^{\tau})]_{\eta(j_{l_i})})$, for all $j_s \in \mathcal{N}(i)$. Hence, by definition of $h_{c,i}$ in (4.10), $u_i(\tau) \in h_{c,i}(x_{\mathbf{q}}^{\tau}) \neq \emptyset$, for all $i \in [1; N]$ from which, condition (2.9) holds for $t = \tau$ as requested in (T2). By (4.18) and Definition 3.1 we get:

$$(4.22) \quad x(\tau) \xrightarrow{u(\tau)} x(\tau + 1).$$

By (4.19) and (4.21) we get

$$\xi_i(\tau) \xrightarrow[\eta,i]{(v_i(\tau), u_i(\tau))} \xi_i(\tau + 1), i \in [1; N]$$

which by (4.17), (4.20) and Definition 3.3 implies

$$(4.23) \quad \xi(\tau) \xrightarrow[\eta]{u(\tau)} \xi(\tau + 1).$$

By (4.22), (4.23), determinism of $S^{\eta}(\Sigma)$, and definition of \mathcal{R}_{μ} , we get (T2.1). Moreover, set $q_{\tau+1} = H_{\mathbf{q}}(x_{\mathbf{q}}^{\tau+1})$. By (T2.1) and definition of $\xi(\tau + 1)$ in (4.20) we get

$$\begin{aligned} \|x(\tau + 1) - \xi(\tau + 1)\| &\leq \mu; \\ \|\xi(\tau + 1) - q_{\tau+1}\| &= \|\xi(\tau + 1) - H_{\mathbf{q}}(x_{\mathbf{q}}^{\tau+1})\| \\ &= \max_{i \in [1; N]} \|\xi_i(\tau + 1) - H_{\mathbf{q},i}(x_{\mathbf{q}}^{\tau+1})\| \\ &\leq \max_{i \in [1; N]} \eta(i)/2 = \|\eta\|/2, \end{aligned}$$

which, combined with (4.11) yields:

$$(4.24) \quad \begin{aligned} \|x(\tau + 1) - q_{\tau+1}\| &\leq \|x(\tau + 1) - \xi(\tau + 1)\| + \|\xi(\tau + 1) - q_{\tau+1}\| \\ &\leq \mu + \|\eta\|/2 \leq \theta. \end{aligned}$$

The above inequality combined with (H3) implies (T2.3). Thus, (T2) is proven.

In order to conclude the proof we need to show that there exists a time $t_f \in \mathbb{N}$ such that $x(t_f) \in \mathcal{X}_f$. Since

$S_{\mathbf{q}}$ is nonblocking there exists a time $t_f \in \mathbb{N}$ such that $x_{\mathbf{q}}^{t_f} \in X_{\mathbf{q},m}$ which implies by (H1) and (4.9) that $x(t_f) \in \mathcal{X}_f$. \square

By Theorem 4.6, definition of $\text{Trim}(S_{Q,\eta})$ and Definition 2.2, it is readily seen that:

Corollary 4.7. *Suppose that Assumption 3.4 holds and select η as required in Theorem 4.6. Then, for any word $\mathbf{q} \in \mathcal{L}_m^y(\text{Trim}(S_{Q,\eta}))$ there exists a triplet $(C, \mathcal{X}_0, \mathcal{X}_f)$ enforcing it on Σ , within accuracy θ .*

By the proof of Theorem 4.6 (see the inequalities (4.13) and (4.24)) the following result holds:

Corollary 4.8. *Suppose that Assumption 3.4 holds. For any desired accuracy $\theta \in \mathbb{R}^+$ select $\mu \in \mathbb{R}^+$ and $\eta \in \mathbb{R}_N^+$ satisfying (3.3) and*

$$(4.25) \quad \mu \leq \theta.$$

If $Y_Q \subset X^\eta$ then sets \mathcal{X}_0 and \mathcal{X}_f in (4.9) and controllers C_i in (2.8) specified by (4.10) solve Problem 2.1.

We conclude this section by discussing the choice in the class of controllers C_i in (2.8).

Remark 4.9. The class of local controllers C_i in (2.8) and specified by (4.10), shown in Theorem 4.6 to solve Problem 2.1, comes out from the general class of specifications we consider and shares analogies with the theory of supervisory control, see also Remark 5.6. When the word $\mathbf{q} = q_0q_1\dots q_{t_f}$ used in (4.8) to define system $S_{\mathbf{q}}$ satisfies the following property

$$(4.26) \quad q_t = q_{t'} \Rightarrow q_{t+1} = q_{t'+1}, \forall t, t' \in [0; t_f - 1], t_f \geq 1,$$

it is possible to show by a slight modification of the proof of Theorem 4.6 that dynamic and open-loop local controllers C_i can be replaced by static local state feedback controllers C'_i in the form of

$$(4.27) \quad u_i \in C'_i(x_i, x_{j_1}, \dots, x_{j_{l_i}}), j_s \in \mathcal{N}(i), i \in [1; N],$$

as often assumed in decentralized control of dynamical systems, where the partial maps $C'_i : \mathbb{R}^{n_i} \times (\times_{j \in \mathcal{N}(i)} \mathbb{R}^{n_j}) \rightarrow 2^{\mathcal{U}_i}$ in (4.27) are specified for any state $\mathbf{x}_{\mathbf{q}}$ of system $S_{\mathbf{q}}$ in (4.8) by:

$$\begin{aligned} (x_1, \dots, x_N) &\in \mathcal{R}_\mu^{-1}([H_{\mathbf{q},1}(\mathbf{x}_{\mathbf{q}})]_{\eta(1)}, \dots, [H_{\mathbf{q},N}(\mathbf{x}_{\mathbf{q}})]_{\eta(N)}), \\ [H_{\mathbf{q},i}(\mathbf{x}_{\mathbf{q}})]_{\eta(i)} &\xrightarrow[\eta,i]{(v_i, u_i)} [H_{\mathbf{q},i}(\mathbf{x}_{\mathbf{q}}^+)]_{\eta(i)}, i \in [1; N], \end{aligned}$$

where $v_i = ([H_{\mathbf{q},j_1}(\mathbf{x}_{\mathbf{q}})]_{\eta(j_1)}, \dots, [H_{\mathbf{q},j_{l_i}}(\mathbf{x}_{\mathbf{q}})]_{\eta(j_{l_i})})$, $j_s \in \mathcal{N}(i)$, $i \in [1; N]$. When instead, word \mathbf{q} violates condition (4.26), the class of controllers C'_i is not general enough for enforcing \mathbf{q} because if $q_t = q_{t'}$ with $t \neq t'$ and $q_{t+1} \neq q_{t'+1}$, controllers C'_i need to enforce transition from q_t to q_{t+1} at time t , and transition from $q_{t'} = q_t$ to $q_{t'+1} \neq q_{t+1}$ at time $t' \neq t$.

5. COMPARISON WITH CENTRALIZED CONTROL ARCHITECTURES

In this section we establish connections with centralized control architectures. A centralized controller for Σ is specified by the dynamic open-loop controller:

$$(5.1) \quad C_c : \begin{cases} x_c(t+1) \in f_c(x_c(t)), \\ u(t) \in h_c(x_c(t)) \subseteq \mathcal{U}, \\ x_c(0) \in X_c^0, \\ x_c(t) \in X_c, t \in \mathbb{N}_0, \end{cases}$$

where $x_c(t)$ is the state of C_c and $u(t)$ is the output of C_c at time t . While state evolution of C_c in (5.1) is nondeterministic, state evolution of C_i in (2.8) is deterministic. This is a consequence of the fact that local controllers C_i need to agree in advance of which word of the specification to enforce. We denote by Σ^{C_c} the control system obtained as coupling Eqns. (2.2) and (5.1). Problem 2.1 rewrites in a centralized setting as:

Problem 5.1. Given Σ in (2.2), L_Q in (2.5) and $\theta \in \mathbb{R}^+$, find $\mathcal{X}_{0,c} \subseteq \mathbb{R}^n$, $\mathcal{X}_{f,c} \subseteq \mathbb{R}^n$ and C_c in (5.1) such that:

- (i) for any trajectory $x(\cdot)$ of Σ^{C_c} as in (2.3) with $x(0) \in \mathcal{X}_{0,c}$, either $x(t_f) \in \mathcal{X}_{f,c}$ or there exists a continuation $x'(\cdot)$ of $x(\cdot)$, as in (2.4), such that $x'(t'_f) \in \mathcal{X}_{f,c}$;
- (ii) for any trajectory $x(\cdot)$ of Σ^{C_c} as in (2.3) with $x(0) \in \mathcal{X}_{0,c}$ and $x(t_f) \in \mathcal{X}_{f,c}$, there exists a word $q_0 q_1 \dots q_{t_f} \in L_Q$ such that for all times $t \in [0; t_f]$ condition (2.10) holds.

For later purposes, we need the following

Definition 5.2. Language $\mathbf{L}_Q(\Sigma^{C_c})$ is the collection of all words $q_0 q_1 \dots q_{t_f} \in L_Q$ for which there exists a trajectory $x(\cdot)$ of Σ^{C_c} as in (2.3) with $x(0) \in \mathcal{X}_{0,c}$ and $x(t_f) \in \mathcal{X}_{f,c}$ satisfying (2.10) for all times $t \in [0; t_f]$.

By the above definition, $\mathbf{L}_Q(\Sigma^{C_c})$ represents the part of L_Q that can be enforced on Σ by C_c . The solution to Problem 5.1 mimicks the one given for the decentralized case. Consider

$$\mathcal{I}_c : (\xrightarrow{Q}) \times \mathbb{R}_N^+ \rightarrow \{\text{True}, \text{False}\}.$$

For any transition $x_Q \xrightarrow{Q} x_Q^+$ of system S_Q defined in the previous section

$$(5.2) \quad \mathcal{I}_c(x_Q \xrightarrow{Q} x_Q^+, \eta) = \text{True}$$

if there exists $u = (u_1, u_2, \dots, u_N) \in U^\eta$ such that conditions (4.3) with $i \in [1; N]$ are jointly satisfied, and $\mathcal{I}_c(x_Q \xrightarrow{Q} x_Q^+, \eta) = \text{False}$, otherwise. Define the subsystem

$$(5.3) \quad S_{Q,\eta}^c = (X_Q^c, X_Q^{0,c}, U_Q^c, \xrightarrow{Q,\eta,c}, X_{Q,m,c}, Y_Q^c, H_Q^c)$$

of S_Q , where $\xrightarrow{Q,\eta,c} \subseteq \xrightarrow{Q}$ contains all and only transitions $x_Q \xrightarrow{Q} x_Q^+$ of S_Q satisfying (5.2). Define

$$(5.4) \quad \text{Trim}(S_{Q,\eta}^c) = (X_T, X_{T,0}, U_T, \xrightarrow{T}, X_{T,m}, Y_T, H_T),$$

and the following sets:

$$(5.5) \quad \begin{aligned} \mathcal{X}_0^c &= \mathcal{R}_\mu^{-1}(\bigcup_{x_T \in X_{T,0}} \times_{i \in [1; N]} \{[H_{T,i}(x_T)]_{\eta(i)}\}), \\ \mathcal{X}_f^c &= \mathcal{R}_\mu^{-1}(\bigcup_{x_T \in X_{T,m}} \times_{i \in [1; N]} \{[H_{T,i}(x_T)]_{\eta(i)}\}). \end{aligned}$$

For any $i \in [1; N]$, function

$$H_{T,i} : X_T \rightarrow \mathbb{R}^{n_i}$$

denotes the "projection" of function H_T onto \mathbb{R}^{n_i} , i.e. for all $x_T \in X_T$, $H_{T,i}(x_T) = q^i$ if $H_T(x_T) = (q^1, q^2, \dots, q^N)$. Entities defining controller C_c in (5.1) are then specified by:

$$(5.6) \quad \begin{aligned} X_c^0 &= X_{T,0}, \\ X_c &= X_T, \\ f_c(x_T) &= \{x_T^+ \in X_T \mid x_T \xrightarrow{T} x_T^+\}, \\ h_c(x_T) &= \left\{ \begin{array}{l} u = (u_1, u_2, \dots, u_N) \in U^\eta \mid \exists x_T^+ \in f_c(x_T) \text{ s.t.} \\ [H_{T,i}(x_T)]_{\eta(i)} \xrightarrow[\eta,i]{(v_i, u_i)} [H_{T,i}(x_T^+)]_{\eta(i)}, i \in [1; N] \end{array} \right\}, \end{aligned}$$

where $v_i = ([H_{T,j_1}(x_T)]_{\eta(j_1)}, \dots, [H_{T,j_i}(x_T)]_{\eta(j_i)})$ with $j_s \in \mathcal{N}(i)$. The following result holds.

Theorem 5.3. Suppose that Assumption 3.4 holds. For any desired accuracy $\theta \in \mathbb{R}^+$ select $\mu \in \mathbb{R}^+$ and $\eta \in \mathbb{R}_N^+$ satisfying (3.3) and (4.11). Then, sets \mathcal{X}_0^c and \mathcal{X}_f^c in (5.5) and controller C_c in (5.1) specified by (5.6) solve Problem 5.1.

The proof of the above result follows the same reasoning as the proof of Theorem 4.6 and is therefore omitted. From the above result, it is readily seen that

Corollary 5.4. *Suppose that Assumption 3.4 holds and select η as required in Theorem 5.3. Then, $\mathbf{LQ}(\Sigma^{C_c}) = \mathcal{L}_m^y(\text{Trim}(S_{Q,\eta}^c))$.*

A direct consequence of Theorem 5.3 and Corollary 5.4 is the following

Corollary 5.5. *Suppose that Assumption 3.4 holds and select η as required in Theorem 5.3. Then, there exists a controller C^c as in (5.1) such that*

$$(5.7) \quad \mathbf{LQ}(\Sigma^{C^c}) = L_Q$$

if and only if

$$(5.8) \quad \text{Trim}(S_Q) = \text{Trim}(S_{Q,\eta}^c).$$

Moreover, if condition (5.8) holds, then C^c in (5.1) specified by (5.6) is such that condition (5.7) holds.

Proof. By Corollary 5.4 and since $\text{Trim}(S_Q) = S_Q$ we get $\mathbf{LQ}(\Sigma^{C^c}) = \mathcal{L}_m^y(\text{Trim}(S_{Q,\eta}^c)) = \mathcal{L}_m^y(\text{Trim}(S_Q)) = \mathcal{L}_m^y(S_Q) = \mathcal{L}_m^u(S_Q') = L_Q$. The second part of the proof holds as a consequence of Theorem 5.3 and Corollary 5.4. \square

Remark 5.6. Corollary 5.5 states that a necessary and sufficient condition for the control system Σ to implement the whole specification L_Q up to a given accuracy θ , is that the specification L_Q is contained in the behavior of the control system Σ , up to the accuracy θ . This result can be viewed as the counterpart in our setting, of the so-called nonblocking controllability theorem (NCT) in the theory of supervisory control of DES, see e.g. [6], establishing sufficient and necessary conditions for the existence of a controller enforcing a regular language specification and such that controlled plant is nonblocking. In particular, *the controller solving the NCT is shown to be any DES marking the specification*; interaction between the plant and the controller is formalized through the notion of parallel composition, where *the controller does not have information on the current state of the plant*. Analogies with the results reported above in this section are noticeable. Indeed, controller C_c replicates the part of the specification system S_Q which can be enforced by Σ and is open-loop.

We conclude this section by establishing connections between the decentralized and centralized control architectures that we proposed. The following result holds.

Theorem 5.7. *Suppose that Assumption 3.4 holds and select η as required in Theorem 4.6 (or equivalently, as required in Theorem 5.3). Then, for any word $\mathbf{q} \in \mathbf{LQ}(\Sigma^{C_c})$ there exists a triplet $(C, \mathcal{X}_0, \mathcal{X}_f)$ enforcing it within accuracy θ .*

Proof. We start by showing $\xrightarrow{Q,\eta} = \xrightarrow{Q,\eta,c}$. Consider any transition $x_Q \xrightarrow{Q,\eta,c} x_Q^+$. By definition of $S_{Q,\eta}^c$ there exists $u = (u_1, u_2, \dots, u_N) \in U^\eta$ such that condition (4.3) holds for all $i \in [1; N]$ from which, $x_Q \xrightarrow{Q,\eta,c} x_Q^+$ is a transition of $S_{Q,\eta}$. Conversely, consider any $x_Q \xrightarrow{Q,\eta} x_Q^+$. Hence, for all $i \in [1; N]$ there exists $u_i \in U_i^\eta$ such that condition (4.3) holds for all $i \in [1; N]$, implying by definition of $\xrightarrow{Q,\eta,c}$ that $x_Q \xrightarrow{Q,\eta,c} x_Q^+$ is a transition of $S_{Q,\eta}^c$. Thus, $\xrightarrow{Q,\eta} = \xrightarrow{Q,\eta,c}$ from which, $S_{Q,\eta} = S_{Q,\eta}^c$ and hence, $\text{Trim}(S_{Q,\eta}^c) = \text{Trim}(S_{Q,\eta})$ which, by Corollaries 4.7 and 5.4, implies the statement. \square

This result shows that any word of the specification that can be enforced by the centralized controller C_c can also be enforced by the decentralized controller C . The only difference is that local controllers C_i need to agree in advance on which word to enforce since in a decentralized control architecture no communication among local controllers is allowed (see Remark 4.3).

6. EFFICIENT CONTROLLERS SYNTHESIS AND COMPUTATIONAL COMPLEXITY ANALYSIS

In this section we extend on-the-fly algorithms of [21] to the synthesis of the decentralized controllers designed in Section 4. The on-the-fly procedure is reported in Algorithm 1, where the main idea is to design controllers C_i in (2.8) without computing explicitly systems $S^\eta(\Sigma_i)$. Starting from system S_Q in (4.1), associated with the specification L_Q , Algorithm 1 returns as output, system $\text{Trim}(S_{Q,\eta})$ and functions $h_{c,i}$, $i \in [1; N]$, on the basis of which, solution to Problem 2.1 is specified in (4.9) and (4.10). It computes in line 6, for each transition $x_Q \xrightarrow{Q} x_Q^+$ and for each control system Σ_i , the set of control inputs $h_{c,i}(x_Q)$. If $h_{c,i}(x_Q) \neq \emptyset$, then transition $x_Q \xrightarrow{Q} x_Q^+$ can be matched by Σ_i by picking any control input $u_i \in h_{c,i}(x_Q)$; in this case, $\mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta)$ is set in line 8 to **True**. If each control system Σ_i can match transition $x_Q \xrightarrow{Q} x_Q^+$, then resulting $\mathcal{I}(x_Q \xrightarrow{Q} x_Q^+, \eta)$ in line 11 evaluates as **True**. System $S_{Q,\eta}$ can then be computed in line 13, by (4.6) that uses \mathcal{I} . $\text{Trim}(S_{Q,\eta})$ is finally computed in line 14. Formal correctness of Algorithm 1 follows from the definition of the sets and operators involved. We also report Algorithm 2 for designing centralized controllers in Section 5, which follows the same reasoning of Algorithm 1.

We conclude with a computational complexity analysis. Let N_Q and $N_{U,i}$ be the cardinality of \xrightarrow{Q} and of \mathcal{U}_i , respectively. It is readily seen that:

Proposition 6.1. *Space and time computational complexity $cc.dec$ of Algorithm 1 scales as $O(N_Q \sum_{i \in [1; N]} N_{U,i})$.*

Proposition 6.2. *Space and time computational complexity $cc.cen$ of Algorithm 2 scales as $O(N_Q \prod_{i \in [1; N]} N_{U,i})$.*

As a direct consequence of Propositions 6.1 and 6.2, when N_Q does not depend on N , as for example in the case of motion planning types specifications, we get

$$cc.cen \sim O(2^N), \quad cc.dec \sim O(N),$$

i.e. from *exponential* complexity with N in the centralized case, to *linear* complexity with N in the decentralized case.

Remark 6.3. It is easy to see that lines 4–10 of Algorithm 1, corresponding to evaluate indicators $\mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta)$ for all $i \in [1; N]$, can be implemented via N computing units, which can work independently from each other, thus leading to a parallel computing architecture. This architecture allows reduction of the time computational complexity bound in Proposition 6.1 from $O(N_Q \sum_{i \in [1; N]} N_{U,i})$ to $O(N_Q \max_{i \in [1; N]} N_{U,i})$ which, when N_Q is independent from N , yields a time computational complexity which is independent from N . Space computational complexity does not reduce in this case and hence, scales as $O(N)$.

In conclusion, centralized and decentralized control architectures allow enforcing the same part of the specification L_Q , with the disadvantage in the decentralized case to agree in advance on which word to enforce, but with advantages in terms of computational complexity.

7. AN ILLUSTRATIVE EXAMPLE

We consider the problem of regulating the temperature in a circular building composed of $N \geq 3$ rooms, each one equipped with a heater. This example set-up is adapted from [13]. The evolution in time of the temperature $\mathbf{T}_i(t)$ of room i with $i \in [1; N]$ is described by control systems Σ_i :

$$(7.1) \quad \begin{aligned} \mathbf{T}_i(t+1) &= \mathbf{T}_i(t) + \alpha(\mathbf{T}_{i+1}(t) + \mathbf{T}_{i-1}(t) - 2\mathbf{T}_i(t)) \\ &\quad + \beta(T_e - \mathbf{T}_i(t)) + \gamma(T_h - \mathbf{T}_i(t))\mathbf{u}_i(t), \end{aligned}$$

where $\mathbf{T}_{i+1}(t)$ and $\mathbf{T}_{i-1}(t)$ are the temperature in Celsius degrees at (discrete) time t of rooms $i+1$ and $i-1$, respectively (here and in the sequel indices 0 and $N+1$ correspond to N and 1, respectively); T_e is the temperature of the external environment of the building; T_h is the temperature of the heater; $\alpha \in \mathbb{R}^+$ is the conduction factor between rooms $i \pm 1$ and room i ; $\beta \in \mathbb{R}^+$ is the conduction factor between the external

```

1 input:  $S_Q = (X_Q, X_{Q,0}, U_Q, \xrightarrow{Q}, X_{Q,m}, \mathbb{R}^n, H_Q)$ ;
2 foreach  $x_Q \xrightarrow{Q} x_Q^+$  do
3   set  $\mathcal{I}(x_Q \xrightarrow{Q} x_Q^+, \eta) := \text{True}$ ;
4   foreach  $i \in [1; N]$  do
5     set  $\mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta) := \text{False}$ ;
6     compute the set  $h_{c,i}(x_Q)$  of all  $u_i \in U_i^\eta$  satisfying (4.3);
7     if  $h_{c,i}(x_Q) \neq \emptyset$  then
8       set  $\mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta) := \text{True}$ ;
9     end
10  end
11  set  $\mathcal{I}(x_Q \xrightarrow{Q} x_Q^+, \eta) := \bigwedge_{i \in [1; N]} \mathcal{I}_i(x_Q \xrightarrow{Q} x_Q^+, \eta)$ ;
12 end
13 compute  $S_{Q,\eta}$  in (4.6);
14 compute  $\text{Trim}(S_{Q,\eta})$ ;
15 output:  $\text{Trim}(S_{Q,\eta})$  and  $h_{c,i}$ ,  $i \in [1; N]$ ;

```

Algorithm 1: Decentralized local controllers design.

```

1 input:  $S_Q = (X_Q, X_{Q,0}, U_Q, \xrightarrow{Q}, X_{Q,m}, \mathbb{R}^n, H_Q)$ ;
2 foreach  $x_Q \xrightarrow{Q} x_Q^+$  do
3   set  $\mathcal{I}_c(x_Q \xrightarrow{Q} x_Q^+, \eta) := \text{False}$ ;
4   compute the set  $h_c(x_Q)$  of all  $u = (u_1, u_2, \dots, u_N) \in U^\eta$  satisfying (4.3) for all  $i \in [1; N]$ ;
5   if  $h_c(x_Q) \neq \emptyset$  then
6     set  $\mathcal{I}_c(x_Q \xrightarrow{Q} x_Q^+, \eta) := \text{True}$ ;
7   end
8 end
9 compute  $S_{Q,\eta}^c$  in (5.3);
10 compute  $\text{Trim}(S_{Q,\eta}^c)$ ;
11 output:  $\text{Trim}(S_{Q,\eta}^c)$  and  $h_c$ ;

```

Algorithm 2: Centralized controller design.

environment and room i ; $\gamma \in \mathbb{R}^+$ is the conduction factor between the heater and room i . Control inputs $\mathbf{u}_i(t)$ at time t assume values in $\mathcal{U}_i = (0.025\mathbb{Z}) \cap [0, 1]$. The specification requires $\mathbf{T}_i(t)$ to follow Table 1 up to an accuracy $\theta = 0.5$. We start by checking Assumption 3.4. Define $A = \max\{|1 - 2\alpha - \beta - \gamma|, |1 - 2\alpha - \beta|\} + 2\alpha$. Network of control systems Σ_i in (7.1) admits the following δ -GAS Lyapunov function

$$(7.2) \quad V(x, x') = \|x - x'\|,$$

for any $x = (\mathbf{T}_1, \mathbf{T}_2, \dots, \mathbf{T}_N) \in \mathbb{R}^N$, $x' = (\mathbf{T}'_1, \mathbf{T}'_2, \dots, \mathbf{T}'_N) \in \mathbb{R}^N$, with \mathcal{K}_∞ functions

$$(7.3) \quad \underline{\alpha}(s) = \overline{\alpha}(s) = s, \quad \rho(s) = (1 - A)s, \quad s \in \mathbb{R}_0^+,$$

provided that

$$(7.4) \quad A < 1.$$

Bounding function σ of V as in (3.2) can be chosen as $\sigma(s) = s$, $s \in \mathbb{R}_0^+$. In this example we pick a uniform quantization $\eta \in \mathbb{R}_N^+$, i.e. $\eta(i) = \eta(j)$ for all $i, j \in [1; N]$ and for ease of notation we use η instead of $\eta(i)$ or $\|\eta\|$. Inequality in (3.3) rewrites as $\eta \leq \mu \min\{(1 - A), 1\}$. While in concrete applications, parameters α , β

and γ need to be identified, in the sequel we choose $\alpha = 0.45$, $\beta = 0.045$, $\gamma = 0.09$, corresponding to the Euler discretization of the model proposed in [13] with sampling time $\tau = 9$. (Larger values of the sampling time lead to instability of the discretized system.) We further set $T_h = 50$ and $T_e = -1$ (L'Aquila is a cold city). We get $A = 0.955$ that satisfies (7.4) so that Assumption 3.4 holds. Specification in Table 1 can be formalized by means of the regular expression \mathbf{qq}^* where

$$(7.5) \quad \mathbf{q} = \begin{aligned} &(19, 18, \dots, 18)(19, 18.5, \dots, 18.5)(19, 19, \dots, 19) \\ &(19, 19.5, \dots, 19.5)(19, 20, \dots, 20)(19, 20, \dots, 20) \\ &(19, 20, \dots, 20)(19, 19.5, \dots, 19.5)(19, 19, \dots, 19) \\ &(20, 18.5, \dots, 18.5)(20, 18.25, \dots, 18.25)(19, 18, \dots, 18). \end{aligned}$$

We get $Y_Q = \{18, 18.25, 18.5, 19, 19.5, 20\}$. For the desired accuracy $\theta = 0.5$ we can pick $\mu = 0.5$ and also $\eta = 0.0225$ which satisfy (3.3) and (4.25). By this choice of η , we get $Y_Q \subset X^\eta$ by which, we can apply Corollary 4.8. Algorithm 1 returns local controllers C_i in Table 2 and $\text{Trim}(S_{Q,\eta}) = S_Q$ where S_Q marks \mathbf{qq}^* . Controllers C_i for $i \in [2; N]$ are of two types: controllers $C_2 = C_N$ and controllers C_i , $i \in [3; N-1]$ that correspond to rooms with neighboring rooms requested to follow different temperature schedules, see Table 1. Sets \mathcal{X}_0 and \mathcal{X}_f involved in Problem 2.1 result in:

$$(7.6) \quad \mathcal{X}_0 = \mathcal{X}_f = [18.5, 19.5] \times \left(\bigtimes_{i \in [2; N]} [17.5, 18.5] \right).$$

Table 2 and (7.6) fully specify the solution to Problem 2.1 which has been solved for an arbitrarily large number N of rooms. We report in Table 3 the results of the simulations on the controlled system. By comparing Tables 1 and 3, and by recalling the accuracy $\theta = 0.5$ chosen, it is readily seen that the specification is met. Time of computation of Algorithm 1 is 0.1563s, without using parallel computing architectures. We solved the same problem for the case of only $N = 4$ rooms by using the centralized approach in Algorithm 2. We obtained $\text{Trim}(S_{Q,\eta}^c) = \text{Trim}(S_{Q,\eta})$, in accordance with Theorem 5.7. Time of computation is 163.6304s. Computations have been performed on a Lenovo IP YOGA 3 PRO 8GB 512SSD.

8. CONCLUSIONS

In this paper we proposed decentralized control architectures for enforcing regular language specifications on networks of discrete-time nonlinear control systems, within any desired accuracy. The approach taken was based on the use of symbolic models and on-the-fly inspired algorithms. A comparison with centralized control architectures was formally discussed which included also a computational complexity analysis. An illustrative example was presented, which showed the applicability and effectiveness of our results.

Acknowledgement: We thank Luca Schenato for fruitful inputs on decentralized control of dynamical systems.

REFERENCES

- [1] R. Alur, Thomas A. Henzinger, Gerardo Lafferriere, and George J. Pappas. Discrete abstractions of hybrid systems. *Proceedings of the IEEE*, 88:971–984, 2000.
- [2] D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.
- [3] A. Borri, D.V. Dimarogonas, K.H. Johansson, M.D. Di Benedetto, and G. Pola. Decentralized symbolic control of interconnected systems with application to vehicle platooning. In *4th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, 2013.
- [4] A. Borri, G. Pola, and M.D. Di Benedetto. Symbolic control design of nonlinear networked control systems. 2016. Submitted for publication. Arxiv 1404.0237.
- [5] F. Bullo, J. Cortés, and S. Martinez. *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.
- [6] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [7] Dan Caugherty. JFLAP: An interactive formal languages and automata package, 1990. Available online at <http://www.jflap.org/>.
- [8] E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

$t \bmod(12)$	$\mathbf{T}_1(t)$	$\mathbf{T}_i(t), i \in [2; N]$
0	19	18
1	19	18.5
2	19	19
3	19	19.5
4	19	20
5	19	20
6	19	20
7	19	19.5
8	19	19
9	19	18.5
10	19	18.25
11	19	18

TABLE 1. Specification L_Q .

$t \bmod(12)$	C_1	$C_2 = C_N$	$C_i, i \in [3; N-1]$
0	{0.65}	{0.45}	{0.6}
1	{0.475}	{0.55}	{0.625}
2	{0.325}	{0.65}	{0.65}
3	{0.15}	{0.75}	{0.65}
4	{0}	{0.525}	{0.35}
5	{0}	{0.525}	{0.35}
7	{0}	{0.175}	{0.025}
8	{0.15}	{0.1}	{0}
9	{0.325}	{0}	{0}
6	{0.475}	{0.075}	{0.15}
10	{0.55}	{0.025}	{0.15}
11	{0.65}	{0.15}	{0.30}

TABLE 2. Local controllers C_i .

$t \bmod(12)$	$\mathbf{T}_1(t)$	$\mathbf{T}_2(t) = \mathbf{T}_N(t)$	$\mathbf{T}_i(t), i \in [3; N-1]$
0	19.5000	18.5000	17.5000
1	18.9788	18.8462	18.0125
2	18.7329	19.2453	18.5368
3	18.6496	19.6773	19.0709
4	18.6042	20.1282	19.5744
5	18.5992	20.1021	19.5924
6	18.6058	20.0838	19.6098
7	18.6176	19.5475	19.1325
8	18.6200	19.0492	18.6292
9	18.6440	18.5357	18.1385
10	18.6448	18.2824	17.8990
11	18.6431	18.0186	17.9080

TABLE 3. Simulation results.

- [9] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory-efficient algorithms for the verification of temporal properties. *Formal Methods in System Design*, 1(2-3):275–288, 1992.
- [10] S.N. Dashkovskiy, H. Ito, and F. Wirth. On a small gain theorem for ISS networks in dissipative Lyapunov form. *European Journal of Control*, 17(4):357–365, 2011.
- [11] E.J. Davison. Decentralized stabilization and regulation in large multivariable systems. In *Directions in Large-Scale Systems*, pages 303–323, USA, 1976. Springer.
- [12] S. Engell. Cyber-physical systems of systems – definition and core research and development areas. Working paper on CPSoS, 2014.
- [13] A. Girard, G. Goessler, and S. Mouelhi. Safety controller synthesis for incrementally stable switched systems using multiscale symbolic models. *IEEE Transactions on Automatic Control*, 2015. In Press. Available at IEEEExplore.org.
- [14] A. Girard and G.J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.
- [15] A. Jadbabaie and J. Lin. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control*, 48(6):988–1001, 2003.
- [16] F. Lin, M. Fardad, and M. R. Jovanovic. Design of optimal sparse feedback gains via the alternating direction method of multipliers. *IEEE Transactions on Automatic Control*, 58(9):2426–2431, 2013.
- [17] Mehran Mesbahi and Magnus Egerstedt. *Graph theoretic methods in multiagent networks*. Princeton series in applied mathematics. Princeton University Press, Princeton (N.J.).
- [18] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.
- [19] R. Olfati-Saber, A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.
- [20] K. Perutka. A survey of decentralized adaptive control. In Meng Joo Er, editor, *New Trends in Technologies: Control, Management, Computational Intelligence and Network Systems*, pages 303–323, 2010.
- [21] G. Pola, A. Borri, and M. D. Di Benedetto. Integrated design of symbolic controllers for nonlinear systems. *IEEE Transactions on Automatic Control*, 57(2):534–539, feb. 2012.
- [22] G. Pola, P. Pepe, and M.D. Di Benedetto. On decentralized approximate supervisory control of networks of nonlinear control systems. In *55th IEEE Conference on Decision and Control*, Las Vegas, NV, USA, December 2016. Submitted.
- [23] G. Pola, P. Pepe, and M.D. Di Benedetto. Symbolic models for networks of control systems. *IEEE Transactions on Automatic Control*, 2016. In Press. Available at IEEEExplore.org.
- [24] M. Rotkowitz and S. Lall. A characterization of convex problems in decentralized control. *IEEE Transactions on Automatic Control*, 51(2):274–286, 2006.
- [25] K. Rudie and W. M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Transactions on Automatic Control*, 37(11):1692–1708, November 1992.
- [26] N.R. Sandell, P. Varaiya, M. Athans, and M.G. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, 23(2):108–128, 1978.
- [27] D.D. Siljak. *Decentralized Control of Complex Systems*. Dover Publications, Inc., Mineola, New York, 2012.
- [28] P. Tabuada. An approximate simulation approach to symbolic control. *IEEE Transactions on Automatic Control*, 53(6):1406–1418, 2008.
- [29] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
- [30] S. Tripakis and K. Altisen. On-the-fly controller synthesis for discrete and dense-time systems. In *World Congress on Formal Methods in the Development of Computing Systems*, volume 1708 of *Lecture Notes in Computer Science*, pages 233 – 252. Springer Verlag, Berlin, September 1999.

9. APPENDIX

9.1. Notation. A directed graph \mathcal{G} is a pair $(\mathcal{V}, \mathcal{E})$ where \mathcal{V} is the set of vertices and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. Symbol \wedge denotes the logical conjunction. Given two sets X and Y and relation $\mathcal{R} \subseteq X \times Y$, symbol \mathcal{R}^{-1} denotes the inverse relation of \mathcal{R} , i.e. $\mathcal{R}^{-1} = \{(y, x) \in Y \times X \mid (x, y) \in \mathcal{R}\}$. Given $X' \subseteq X$ and $Y' \subseteq Y$, we denote $\mathcal{R}(X') = \{y \in Y \mid \exists x \in X' \text{ s.t. } (x, y) \in \mathcal{R}\}$ and $\mathcal{R}^{-1}(Y') = \{x \in X \mid \exists y \in Y' \text{ s.t. } (x, y) \in \mathcal{R}\}$. Symbols \mathbb{N}_0 , \mathbb{Z} , \mathbb{R} , \mathbb{R}^+ and \mathbb{R}_0^+ denote the set of nonnegative integer, integer, real, positive real, and nonnegative real numbers, respectively. Symbol \mathbb{R}_n^+ denotes the positive orthant of \mathbb{R}^n . The symbol 0_n denotes the origin of \mathbb{R}^n . Given $n \in \mathbb{N}_0$ and $n > 0$, symbol $[1; n]$ denotes $\{1, 2, \dots, n\}$. Given $x \in \mathbb{R}^n$, symbol $x(i)$ denotes the i -th element of x and $\|x\|$ the infinity norm of x . Given $a \in \mathbb{R}$ and $X \subseteq \mathbb{R}^n$, symbol aX denotes the set $\{y \in \mathbb{R}^n \mid \exists x \in X \text{ s.t. } y = ax\}$. Given $\theta \in \mathbb{R}^+$ and $x \in \mathbb{R}^n$, we define $\mathcal{B}_{[\theta]}(x) = \{y \in \mathbb{R}^n \mid y(i) \in [x(i) - \theta, x(i) + \theta], i \in [1; n]\}$. Note that for any $\theta \in \mathbb{R}^+$, $\{\mathcal{B}_{[\theta]}(x)\}_{x \in 2\theta \mathbb{Z}^n}$ is a partition of \mathbb{R}^n . Given $z \in \mathbb{R}^n$, symbol $[z]_\theta$ denotes the unique vector in $\theta \mathbb{Z}^n$ such that $z \in \mathcal{B}_{[\theta/2]}([z]_\theta)$. A continuous function $\rho : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is said to belong to class \mathcal{K} if it is strictly increasing and $\rho(0) = 0$; function ρ is said to belong to class \mathcal{K}_∞ if $\rho \in \mathcal{K}$ and $\rho(r) \rightarrow \infty$ as $r \rightarrow \infty$.

9.2. Systems, regular languages and approximate bisimulation. We recall from e.g. [6] some notions on formal language theory. Let Y be a finite set representing the alphabet. A word over Y is a finite sequence $y_1 y_2 \dots y_l$ of symbols in Y . The empty word is denoted by ε . The symbol Y^* denotes the Kleene closure of Y , that is the collection of all words over Y including ε . A language L over Y is a subset of Y^* . We now recall the notion of system:

Definition 9.1. A system is a tuple $S = (X, X_0, U, \longrightarrow, X_m, Y, H)$, consisting of a set of states X , a set of initial states $X_0 \subseteq X$, a set of inputs U , a transition relation $\longrightarrow \subseteq X \times U \times X$, a set of marked states $X_m \subseteq X$, a set of outputs Y and an output function $H : X \rightarrow Y$.

The above definition slightly extends the one of [29] to systems with marked states. A transition $(x, u, x') \in \longrightarrow$ of S is denoted by $x \xrightarrow{u} x'$. System S is empty if $X_0 = \emptyset$. The evolution of systems is captured by the notions of state, input and output runs. Given a sequence of transitions of S

$$(9.1) \quad x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} \dots \xrightarrow{u_{l-1}} x_l$$

with $x_0 \in X_0$, the sequences

$$(9.2) \quad r_X : x_0 x_1 \dots x_l,$$

$$(9.2) \quad r_U : u_0 u_1 \dots u_{l-1},$$

$$(9.3) \quad r_Y : H(x_0) H(x_1) \dots H(x_l),$$

are called a *state run*, an *input run* and an *output run* of S , respectively. System S is said to be *symbolic/finite* if X and U are finite sets, *metric* if Y is equipped with a metric $\mathbf{d} : Y \times Y \rightarrow \mathbb{R}_0^+$, *deterministic* if for any $x \in X$ and $u \in U$ there exists at most one transition $x \xrightarrow{u} x^+$ and *nondeterministic*, otherwise. System S is said *nonblocking* if for any transitions sequence (9.1) of S with $x_0 \in X_0$ either $x_l \in X_m$ or there exists a continuation $x_0 \xrightarrow{u_0} x_1 \xrightarrow{u_1} \dots \xrightarrow{u_{l-1}} x_l \xrightarrow{u_l} \dots \xrightarrow{u_{l'-1}} x_{l'}$ of it such that $x_{l'} \in X_m$, and *blocking*, otherwise. The *input language* (resp. *output language*) of S , denoted $\mathcal{L}^u(S)$ (resp. $\mathcal{L}^y(S)$), is the collection of all its input runs (resp. output runs). The *marked input language* (resp. *marked output language*) of S , denoted as $\mathcal{L}_m^u(S)$ (resp. $\mathcal{L}_m^y(S)$), is the collection of all input runs r_U in (9.2) (resp. output runs r_Y in (9.3)) such that the corresponding transitions sequence in (9.1) is with ending state $x_l \in X_m$. A language L over a finite set U is said *regular* if there exists a symbolic system S with input set U such that $L = \mathcal{L}_m^u(S)$. We also recall some unary operations on systems naturally adapted from the ones given for DES [6]. A system $S' = (X', X'_0, U', \longrightarrow', X'_m, Y', H')$ is said to be a *subsystem* of $S = (X, X_0, U, \longrightarrow, X_m, Y, H)$, denoted $S' \sqsubseteq S$, if $X' \subseteq X$, $X'_0 \subseteq X_0$, $U' \subseteq U$, $\longrightarrow' \subseteq \longrightarrow$, $X'_m \subseteq X_m$, $Y' \subseteq Y$ and $H'(x) = H(x)$ for all $x \in X'$. The accessible part of S , denoted $\text{Ac}(S)$, is the unique maximal² subsystem S' of S such that for any state x' of S' there exists a state run of S' ending in x' . By definition, if S is nonempty, $\text{Ac}(S)$ is accessible. The co-accessible part of S , denoted $\text{Coac}(S)$, is the unique maximal² subsystem S' of S such that for any state $x' \in X'$ there exists a transition sequence of S' starting from x' and ending in a marked state of S' . By definition, $\text{Coac}(S)$ if not empty, is nonblocking. The trim of S , denoted $\text{Trim}(S)$, is defined as $\text{Trim}(S) = \text{Coac}(\text{Ac}(S)) = \text{Ac}(\text{Coac}(S))$. By definition, $\text{Trim}(S)$, if not empty, is accessible and nonblocking. We conclude by recalling some notions related to systems' simulation and bisimulation:

Definition 9.2. [4] Let $S_i = (X_i, X_{0,i}, U_i, \xrightarrow{i}, X_{m,i}, Y_i, H_i)$ ($i = 1, 2$) be metric systems with the same input set $U_1 = U_2$, output sets $Y_1 = Y_2$ and metric \mathbf{d} , and let $\mu \in \mathbb{R}_0^+$ be a given accuracy. A relation $\mathcal{R} \subseteq X_1 \times X_2$ is said a strong μ -approximate simulation relation from S_1 to S_2 if it enjoys the following conditions:

- (i) $\forall x_1 \in X_{0,1} \exists x_2 \in X_{0,2}$ such that $(x_1, x_2) \in \mathcal{R}$;
- (ii) $\forall x_1 \in X_{m,1} \exists x_2 \in X_{m,2}$ such that $(x_1, x_2) \in \mathcal{R}$;
- (iii) $\forall (x_1, x_2) \in \mathcal{R}, \mathbf{d}(H_1(x_1), H_2(x_2)) \leq \mu$;

²Here, maximality is with respect to the pre-order naturally induced by the binary operator \sqsubseteq .

(iv) $\forall (x_1, x_2) \in \mathcal{R}$ if $x_1 \xrightarrow[1]{u} x'_1$ then there exists $x_2 \xrightarrow[2]{u} x'_2$ such that $(x'_1, x'_2) \in \mathcal{R}$.

Relation \mathcal{R} is a strong μ -approximate bisimulation relation between S_1 and S_2 if \mathcal{R} is a strong μ -approximate simulation relation from S_1 to S_2 and \mathcal{R}^{-1} is a strong μ -approximate simulation relation from S_2 to S_1 . Systems S_1 and S_2 are strongly μ -bisimilar, denoted $S_1 \cong_\mu S_2$, if there exists a strong μ -approximate bisimulation relation \mathcal{R} between S_1 and S_2 .

The above notion requires stronger conditions than approximate (bi)simulation of [14] that allows transitions in condition (iv) with possibly different control labels.

¹ DEPARTMENT OF INFORMATION ENGINEERING, COMPUTER SCIENCE AND MATHEMATICS, CENTER OF EXCELLENCE DEWS, UNIVERSITY OF L'AQUILA, 67100 L'AQUILA, ITALY

E-mail address: {giordano.pola,pierdomenico.pepe,mariadomenica.dibenedetto}@univaq.it